

## **Amendments to the Claims**

Claims 1 - 2 (canceled)

1      Claim 3 (previously presented): The method according to Claim 32, wherein the input document  
2      is a structured document.

1      Claim 4 (previously presented): The method according to Claim 3, wherein the structured  
2      document is encoded in Extensible Markup Language (“XML”).

1      Claim 5 (previously presented): The method according to Claim 32, wherein the generated  
2      output comprises at least one object representation generated from the input document.

Claim 6 (canceled)

1      Claim 7 (previously presented): The method according to Claim 33, wherein the second schema  
2      definition is requested by specifying a schema name of the second schema definition, to which the  
3      generated output must adhere.

1      Claim 8 (previously presented): The method according to Claim 33, wherein the second schema  
2      definition is requested by specifying a schema name of the second schema definition, indicating  
3      that the second schema definition is to be used by the validating parser when generating the  
4      output.

1       Claim 9 (previously presented): The method according to Claim 8, wherein the schema name is  
2       specified, by the application program, as a feature on an invocation of the validating parser.

Claim 10 (canceled)

1       Claim 11 (previously presented): The method according to Claim 32, wherein the identification of  
2       the first schema definition in the input document comprises a specification, in the syntax of the  
3       input document, of the first schema definition.

1       Claim 12 (previously presented): The method according to Claim 32, wherein the identification of  
2       the first schema definition in the input document uses a schema location construct in the input  
3       document.

1       Claim 13 (previously presented): A computer-implemented method of casting objects,  
2       comprising:  
3               providing a validating parser that is adapted for validating whether syntax elements of an  
4       input document conform to a first schema definition identified in the input document while  
5       generating output objects, from the validated syntax elements of the input document, that  
6       conform to a second schema definition dynamically selected by a consuming application of the  
7       generated output objects;  
8               using the validating parser for validating whether the syntax elements of the input

9 document conform to the first schema definition, wherein:

10                   the first schema definition is an extended schema;

11                   using the validating parser, responsive to the validating of the syntax elements, for

12                   generating the output objects to conform to the second schema definition, wherein:

13                   the second schema definition is a base schema from which the extended schema

14                   was extended, such that the extended schema defines at least one syntax element that is not

15                   defined in the base schema; and

16                   the generating further comprises not generating any output object for any of the at

17                   least one syntax element that is defined in the extended schema but not defined in the base schema

18                   in order that the generated output objects will conform to the second schema definition; and

19                   providing the generated output objects, by the validating parser, for use by the consuming

20                   application.

Claims 14 - 19 (canceled)

1 Claim 20 (currently amended): The method according to Claim 13, wherein:

2                   ~~the second schema definition is the base schema;~~

3                   an intermediate schema definition extends the base schema by adding at least one syntax

4                   element not defined in the base schema; and

5                   the first schema definition extends the intermediate schema definition by adding at least

6                   one syntax element not defined in the intermediate schema definition; and

7                   the generating further comprises not generating any output object for any of the at least

8 one syntax element that is defined in the intermediate schema but not in the base schema.

Claims 21 - 30 (canceled)

1       Claim 31 (previously presented): A computer-implemented method of providing validation and  
2       parsing, comprising:

3               providing a validating parser adapted for validating an input document according to a first  
4       schema definition identified in the input document while generating output, from the validated  
5       input document, according to a second schema definition dynamically selected by a consuming  
6       application of the generated output;

7               validating syntax elements of the input document with the provided validating parser  
8       according to the first schema definition, wherein the first schema definition is an extended schema  
9       which specifies a syntax definition to which the syntax elements of the input document are to  
10      adhere; and

11               responsive to the validating of the syntax elements, parsing the validated syntax elements  
12      to generate the output for the consuming application according to the second schema definition,  
13      wherein the second schema definition is a base schema from which the extended schema was  
14      extended, thereby suppressing at least one of the validated syntax elements when generating the  
15      output for the consuming application, wherein each of the suppressed syntax elements is valid  
16      according to the extended schema but is not valid according to the base schema.

1       Claim 32 (previously presented): A computer-implemented method of applying abstraction by a

2 validating parser, comprising:

3 using, by a validating parser, a first schema definition for validating syntax elements when  
4 parsing syntax of an input document, wherein the first schema definition is identified in the input  
5 document; and

6 omitting, by the validating parser when generating output from the parsed syntax of the  
7 input document, each of at least one of the validated syntax elements which is valid according to  
8 the first schema definition but is not valid according to a second schema definition for which the  
9 output is generated, wherein:

10 the first schema definition is an extended schema; and

11 the second schema definition is a base schema from which the extended schema is  
12 extended, such that the extended schema defines at least one syntax element that is not defined in  
13 the base schema.

1 Claim 33 (previously presented): The method according to Claim 32, wherein the second schema  
2 definition is dynamically requested, to the validating parser, by an application program for which  
3 the output is being generated.

1 Claim 34 (new): The computer-implemented method according to Claim 31, wherein:

2 an intermediate schema definition extends the base schema by adding at least one syntax  
3 element not defined in the base schema; and  
4 the first schema definition extends the intermediate schema definition by adding at least  
5 one syntax element not defined in the intermediate schema definition; and

the suppressing also suppresses each syntax element that is valid according to the intermediate schema but which is not valid according to the base schema.

Claim 35 (new): The computer-implemented method according to Claim 32, wherein:

- an intermediate schema definition extends the base schema by adding at least one syntax element not defined in the base schema; and
- the first schema definition extends the intermediate schema definition by adding at least one syntax element not defined in the intermediate schema definition; and
- the omitting also omits each validated syntax element that is valid according to the intermediate schema but which is not valid according to the base schema.

Claim 36 (new): A system for applying abstraction with a validating parser, comprising:

- a validating parser usable by a computer; and
- instructions which are executable, using a processor of the computer, to perform:
  - using, by the validating parser, a first schema definition for validating syntax elements when parsing syntax of an input document, wherein the first schema definition is identified in the input document; and
  - omitting, by the validating parser when generating output from the parsed syntax of the input document, each of at least one of the validated syntax elements which is valid according to the first schema definition but is not valid according to a second schema definition for which the output is generated, wherein:
    - the first schema definition is an extended schema; and

12 the second schema definition is a base schema from which the extended  
13 schema is extended, such that the extended schema defines at least one syntax element that is not  
14 defined in the base schema.

1 Claim 37 (new): The system according to Claim 36, wherein the second schema definition is  
2 dynamically requested, to the validating parser, by an application program for which the output is  
3 being generated.

1       Claim 38 (new): The system according to Claim 36, wherein:

2               an intermediate schema definition extends the base schema by adding at least one syntax

3               element not defined in the base schema; and

4               the first schema definition extends the intermediate schema definition by adding at least

5               one syntax element not defined in the intermediate schema definition; and

6               the omitting also omits each validated syntax element that is valid according to the

7               intermediate schema but which is not valid according to the base schema.

1       Claim 39 (new): A computer program product for applying abstraction with a validating parser,  
2       the computer program product embodied on at least one computer-readable medium and  
3       comprising computer-readable program code for:  
4                using, by a validating parser, a first schema definition for validating syntax elements when  
5       parsing syntax of an input document, wherein the first schema definition is identified in the input  
6       document; and

7 omitting, by the validating parser when generating output from the parsed syntax of the  
8 input document, each of at least one of the validated syntax elements which is valid according to  
9 the first schema definition but is not valid according to a second schema definition for which the  
10 output is generated, wherein:

11 the first schema definition is an extended schema; and

12 the second schema definition is a base schema from which the extended schema is  
13 extended, such that the extended schema defines at least one syntax element that is not defined in  
14 the base schema.

1 Claim 40 (new): The computer program product according to Claim 39, wherein the second  
2 schema definition is dynamically requested, to the validating parser, by an application program for  
3 which the output is being generated.

1       Claim 41 (new): The computer program product according to Claim 39, wherein:

2               an intermediate schema definition extends the base schema by adding at least one syntax

3               element not defined in the base schema; and

4               the first schema definition extends the intermediate schema definition by adding at least

5               one syntax element not defined in the intermediate schema definition; and

6               the omitting also omits each validated syntax element that is valid according to the

7               intermediate schema but which is not valid according to the base schema.